

Analysis on C++ Topic Difficulties Ranking: A Case Study on Mechanical Engineering Students in UiTM Pasir Gudang

H. Hasmy, N.S. Ibrahim and A.Y. Kapi

*Faculty of Computer and Mathematical Sciences
Universiti Teknologi MARA, Cawangan Johor, Kampus Pasir Gudang
Jalan Purnama, Bandar Seri Alam, 81750 Masai, Johor, Malaysia*

harshida@uitm.edu.my

Abstract—This paper aims to analyse student's performance in the course "Fundamentals of Computer Problem Solving" (CSC128) final examination results. CSC128 is a C++ programming course, and it is a compulsory subject for all Diploma in Mechanical Engineering and Diploma in Civil Engineering students at Universiti Teknologi MARA Pasir Gudang Campus. Through the analysis, this paper identifies which topics in CSC128 are not well mastered by the students. The data was collected from 163 students score marks and the analysis was conducted by categorising the final examination questions into five different topics according to the CSC128's syllabus. An indicator has been used to classify students' performance for each topic by comparing the percentage of students who scored 50% above and below of the total marks. The study identified that Topic 4, "Repetition Control Structure" was placed in the first rank as the most challenging topic encountered by the students, and Topic 3, "Selection Control Structure" was ranked as the least difficult topic. The findings will be used to improve the subject in order to achieve the course outcomes and can be a guideline for the lecturers to improve their teaching method in order to increase students' understanding, interest, and performance in programming.

Keywords—C++ Computer programming; difficulties; teaching and learning

I. INTRODUCTION

THE necessity for having problem-solving skills among the engineering students - is indisputable. Currently, most of the work areas are related to computers and automation tools. It is crucial for the engineering students to learn fundamental computer programming skills in order to solve problems using computers. The "Fundamentals of Computer Problem Solving" (CSC128) course is an introduction to problem-solving using computers, which highlights various aspects of problemsolving such as defining the phases, planning the solutions, and solve the problems using a C++ programming language. CSC128 is a mandatory course for all fourth-semester students of Diploma in Mechanical Engineering and Diploma in Civil Engineering second semester students at Universiti Teknologi MARA (UiTM) Johor Branch Pasir Gudang Campus. However, teaching programming language to engineering students is quite challenging, and most of the students believe that programming is a difficult subject to score and understand [1].

The main objectives of this study are to analyse the student's performance through the final examination and to identify which topics in CSC128 are not mastered by students. The key interest is to identify the most challenging topic in the syllabus for students in order to provide the most effective teaching method and learning experience for the students in the future to improve their understanding of this course. There are various types of teaching

programming techniques, and practising the correct method would produce better results.

In order to discover the most challenging topic in CSC128, 163 Mechanical Engineering student's final examination scores from March–July 2019 batch have been collected. In this paper, the following section presents a review of previous work in the introductory programming language, and the methodology used in this study is explained in the third section. The content for each course learning outcome will be further discussed in the result and discussion. Results and discussion are presented in the fourth section by discussing the most challenging topic in CSC128 at UiTM Pasir Gudang. The conclusion is in the final section to end the discussion and presents a future recommendation for future work.

II. PREVIOUS WORKS

Computer programming is a course focusing on the process of problem solving, which involves the skill in designing algorithms, writing program, understanding the syntax as well as the logic of the program [1]. In the higher education level, this course is not only compulsory for Computer Sciences' students, but also a part of the curriculum for most of Science and Technology courses.

Even though programming is a typical course for engineering students, teaching and learning programming languages such as C++ is not an easy or simple task At Universiti Teknologi MARA Malaysia, introduction to a programming language (C++) is included in the curriculum for numerous engineering undergraduate students such as Mechanical and Civil. Programming is said to be one of the most challenging courses to master due to various reasons and grouped into three categories; the course itself, students and instructors [1].

Many institutions reported that most of their students found that the programming course is hard to understand and scored [1][2][3]. The main reason that leads to the problems in mastering programming is the difficulty of the course itself. It is burdensome to score in programming course from the

students' point of view since most of them have no basic programming knowledge. Therefore, understanding the basic concepts of programming structure, designing a program as well as studying the language syntax are challenging tasks for them [2].

There have been numerous studies that examined problems and challenges that both students and lecturers faced in dealing with C++. Study in [4] disclosed that there are a few reasons why learning programming is a big challenge to the students. Firstly, students are unable to understand the syntax and semantic of programming as well as weak in logical operations. Therefore, they will have problems in solving the programming questions and manipulating the code of the program. Secondly, the students' learning approach is also one of the main reasons. Their dependency on the learning process solely in the class and lack of practising after the class often lead to difficulties in learning programming.

There are several reasons why students find it challenging to understand programming. As stated in [5], the reasons are as follows: i) the inability to relate the problem with computing and programming knowledge, ii) stunned and panicked with the number of syntax errors published after execution process, iii) unable to define the type of error and fix it due to lack of understanding, iv) incapable of distinguishing the differences of every programming structures such as selection and repetition. In particular, a study in [6] summarised a learner's misinterpretation of concept in syntax, theoretical and critical wisdom. The misinterpretation often leads to low student motivation towards the learning process in most of the students. Furthermore, a reference in [7] reported that instructors of an introductory programming language often encountered problems with student's negative attitude. These attitudes include the inability to finish assignments or tutorial before going to the class or duplicating their classmates' as a desperate solution.

On the instructor's point of view, students need to familiarise with the abstract and syntax through a series of tutorials and assignments.

However, [7] expressed that grading the assignment/tutorial required a tremendous amount of instructor's resources. On the other hand, traditional teaching methods also contribute to the problems. The instructors should improvise their teaching approach by adopting 21st century teaching strategies to attract the students' interest in order to improve programming education [4].

Based on previous studies, it is shown that the teaching and learning of introductory programming language are most difficult. Both the instructors and students often suffered from the challenges in teaching and learning an introductory programming language. Since the introductory programming language is an essential subject in engineering, it becomes the main issue which needs to be addressed.

Many researchers came up with various ideas to boost students' performance and understanding in programming course, especially in improving the teaching methods. In 2018, [8] developed an online quiz application; i-SAQ (C++ Interactive Self-Assessment Quiz) which provide immediate feedback to the students for each question. They conducted a usability test and reported positive feedback from students towards the quiz. Study in [9] introduced several game elements for teaching in a programming course. Results demonstrated a clear distinction between control and experimenting groups which results in student's motivation and effectiveness. Meanwhile, recent research by [10] suggests pair programming as a technique to help students to improve their programming learning process. The technique allows the student to enhance their understanding and skills by producing better quality programming codes in a shorter timescale and helped them in getting better marks from the collaborative and teamwork activities.

Most of the past researches conducted to date cover various solutions for education in teaching and learning an introductory programming language. However, limited research has been done to study the root of the cause and which topic demands special needs from both students and lecturers in teaching and learning introductory programming course

specifically for C++. Therefore, this study is focused on finding the most challenging topic evaluated from student's performance in the final examination at Universiti Teknologi MARA, Pasir Gudang Campus.

III. METHODOLOGY

This study focused on the final examination, which contributed to 50% of the student's grade. The final examination paper questions emphasize to test student's knowledge, their thinking and scientific skills. The total marks for the final examination paper is 100, which need to be answered within three hours. The final examination consists of three parts: Part A, B and C. Part A consisted of 10 multiple choice questions, Part B comprised 19 short answer questions and Part C comprised two types of long essay questions.

Undergraduate students who enrolled in CSC128 from Faculty of Mechanical Engineering has been selected for this study and no prior knowledge of computing is required to enrol in the course. There are 163 students from 7 different classes and has been taught by two different lecturers. In order to analyse the student's performance, the questions were then categorised according to the topics in CSC128 syllabus. There is a total of five topics in the syllabus which are being taught throughout 14 weeks in one semester. The first topic introduced was programming language and described five steps in the Program Development Life Cycle (PDLC). The second topic introduced fundamental elements in programming (C++), including basic data types, arithmetic and assignment statement. The third topic covered types of selection control structure which involves boolean values and expression, relational and logical operators. Repetition control structures introduced in the fourth topic and the final topic covered functions, including function call and parameters passing.

CSC128 final examination comprised of 31 questions that are compulsory for the students, which were split into five topics which are outlined by the CSC128 course, as shown in TABLE 1.

TABLE 1. Mark Distribution in Final Examination by Questions

Part	Question No.	Marks	Topic
A	1	2	1
	2	2	1
	3	2	2
	4	2	2
	5	2	3
	6	2	4
	7	2	4
	8	2	4
	9	2	5
	10	2	5
B	1a	2	2
	1b	4	1
	1c	4	2
	2ai	2	2
	2aii	2	2
	2bi	2	2
	2bii	2	2
	3ai	4	3
	3aii	2	4
	3bi	4	3
	3bii	4	4
	4ai	2	4
	4aii	2	4
	4bi	3	4
	4bii	3	4
5ai	1	5	
5aii	1	5	
5aiii	1	5	
5b	5	5	
C	1	15	1,2,3,4,5
	2	15	1,2,3,4,5

From TABLE 1, it is clearly shown that Part C was comprised of all topics in CSC128, and thus, Part C is eliminated in this study due to dissimilarity from other questions. Besides, the eradication of Part C is required to avoid invalid results for comparison in student's performance sorted by topic. Due to this reason, two questions were excluded from the analysis.

Hence, 29 questions extracted from CSC128 final examination paper which consist of 10 multiple choice questions and 19 short

answer questions and its total marks were 70. All questions were checked and marked through syndicated marking process by five different lecturers. Selected questions were then sorted according to the topics and marks for each topic is recorded, as shown in TABLE 2.

TABLE 2. Distribution of Total Marks by Topics

Topic	Number of questions	Total marks	Mid-value
1. Introduction	3	8	4
2. Basic Elements of a Computer Program	8	18	9
3. Selection Control Structure	3	10	5
4. Repetition Control Structure	9	22	11
5. Functions	6	12	6

In classifying the most challenging topic encountered by the students, an indicator is generated to form a basis in classified marks. In the analysis, the indicator is considered as 50% of full marks for each topic as expressed in TABLE 3.

TABLE 3. Indicators for Classifying Marks

Indicator	Description
Above mid-value	Percentage of students who scored at least half of full marks (above 50% of full marks)
Below mid-value	Percentage of students who got below half of the full marks (below 50% of full marks)

Based on TABLE 3, this indicator plays essential roles in classifying student's performance for each topic in CSC128's syllabus. From the indicator, the percentage of students that have mastered the topics will be collected and sorted accordingly from the highest to the lowest percentage of below mid-value of full marks. The lowest percentage indicate topic that have been mastered by students while the highest specify otherwise.

A COUNTIF function in a spreadsheet is used to count the number of students who scored for each mark and topic. By using the indicator, percentage below and above mid-

value of full marks were tracked for each topic. As a result, this study ranked out the topic in CSC128 that have not been mastered by the students in the scope and it will be discussed in the next section.

IV. RESULT AND DISCUSSION

This study was carried out by analysing the final examination marks for 163 undergraduate students from the Diploma in Mechanical Engineering. There is a total of 31 questions out of five topics under course CSC128 which contribute to 100 marks has been evaluated in previous June 2019 final examination. However, in this research, only 29 questions which contribute to 70 marks are taken into consideration.

As demonstrated previously in TABLE 1, the distribution of several questions and total marks varies among the evaluated topics. The distribution is due to the mapping of Course Learning Outcome (CLO), and Cognitive Level for each topic is differ based on Test Specification Table (JSU). The process of preparing the final exam question paper is based on the JSU prepared by the UiTM's course resource person. Every question must follow the cognitive level in order to meet all three course learning outcomes. The cognitive level is divided into six levels where the first and second level (C1 and C2) is easy, followed by C3 and C4 for moderate difficulties and lastly C5 and C6 with the hardest difficulties. Fig. 1 depicts the distribution of total marks by topics in percentage.

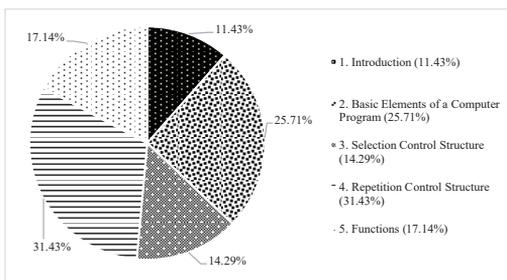


Fig. 1. Percentage Distribution of Total Marks by Topics

From Fig. 1, Topic 1, "Introduction" contributes to the lowest percentage of evaluated marks which is 11.43%. The reason

for this is that Topic 1 is just focusing on theoretical part. Therefore, assessment for this topic only applicable for CLO 1 (Identify the steps and requirements of given problems using a systematic problem-solving approach) and the Cognitive Level evaluated were C1 and C2 (Easy) only.

Contrarily, Topic 4, "Repetition Control Structure" contributes to the highest percentage of evaluated marks which is 31.43%. All three CLOs, which are CLO 1 (Identify the steps and requirements of given problems using systematic problem-solving approach), CLO 2 (Write complete programs using structural and modular approach) and CLO 3 (Demonstrate basic program to solve daily problems using designated programming control structures (selection, repetition and/or function)) were tested for this topic. As for Cognitive Level, Topic 4 is evaluated based on level C1 (easy), C3 and C4 (moderate).

This study focuses on analysing students' performance by identifying the most challenging topic encountered by the students. The indicator that has been used to classify students' performance for each topic is by comparing the percentage of students who scored below and above mid-value of total marks for each topic. The percentage comparison result for five evaluated topics is illustrated in Fig. 2.

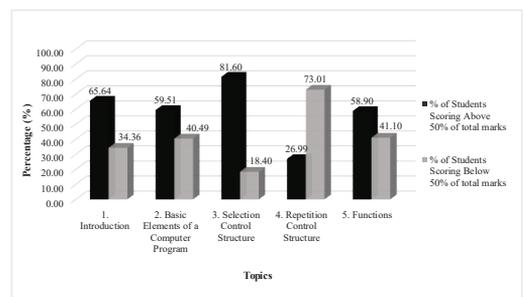


Fig. 2. Percentage Comparison for Students who Scored Below and Above Mid-value of Total Marks for Each Topic

From Fig. 2, only 26.99% out of 163 students able to score above mid-value of the total mark for Topic 4, while another 73.01% of the students scored below the mid-value. Other than Topic 4, the percentage of students who scored at least half of the total mark surpasses

the percentage of students who got below half of the total marks. The massive distinction in percentage for Topic 4 with other topics shows that most of the students incapable of answering questions related to this topic. To give a clear perspective, Fig. 3 illustrated the percentage of students who scored below the mid-value of total marks for each topic.

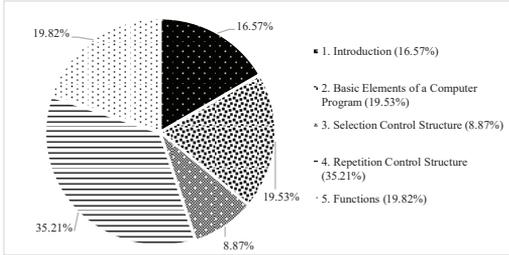


Fig. 3. Percentage of Students Who Scored Below Mid-value of Total Marks for Each Topic

Fig. 3 shows the percentage of students out of 163 students who scored below mid-value of total marks for each topic. Based on the result, Topic 4 demonstrates the highest percentage with 35.21%, followed by Topic 5 with 19.82%. Topic 2 is in the third rank, with 19.53%, followed by Topic 1 with 16.57%. Topic 3 shows the lowest percentage, with 8.87%. The lower percentage of students who scored below half of the total marks indicates that the students have mastered the topic. The ranking of topics in CSC128's syllabus that is difficult to score by the students based on the final examination result is simplified in the following TABLE 4.

As expressed in TABLE 4, this study reveals that Topic 4, "Repetition Control Structure" which is placed in the first rank, is the most challenging topic encountered by the students. Conversely, Topic 3, which is "Selection Control Structure" is shown otherwise. As illustrated in Fig. 1 formerly, Topic 4 contributes to the highest percentage distribution of total marks in CSC128 final examination. Thus, it will have a significant impact on students' overall achievement. Therefore, various actions should be considered to overcome this limitation.

TABLE 4. Topic Difficulty Ranking in CSC128

Ranking	Topic
1	4. Repetition Control Structure
2	5. Functions
3	2. Basic Elements of a Computer Program
4	1. Introduction
5	3. Selection Control Structure

The results obtained from this study can be a guideline for the instructors to seek the appropriate improvement in teaching methods in order to boost up students' understanding and performance. Both students and instructors should also consider expanding Students' Learning Time (SLT) for topic Repetition Control Structure through extra exercises, examples, and assessments.

V. CONCLUSION

In conclusion, this paper has identified the fourth topic from CSC128 syllabus, "Repetition Control Structure" as the most difficult topic with the lowest marks gained by the students in their final examination results. There is 35.21% out of 163 students who scored below half of the total marks. From this analysis, it is pointed out that the repetition control structure is challenging for some students to visualise the concept of a loop. Students were required to have substantial prior knowledge of the previous topic to be able to grasp the concept of the loop in programming.

In the future, solutions should be designed to improve the understanding of students towards Topic 4 in CSC128. The teaching method will be revised to include various teaching tools to help students understand the topic. Besides, further study will focus on teaching tools to attract students' interest and engage them in learning CSC128.

REFERENCES

[1] M. Rahmat, S. Shahrani, R. Latih, N.F.M. Yatim, N.F.A. Zainal, and R.A. Rahman, "Major problems in basic programming that influence student performance," *Procedia-Social Behav. Sci.*, vol. 59, pp. 287–296, 2012.

- [2] A. Baist and A.S. Pamungkas, "Analysis of Student Difficulties in Computer Programming," *VOLT J. Ilm. Pendidik. Tek. Elektro*, vol. 2, no. October, pp. 81–92, 2017.
- [3] K.A. Sarpong, J.K. Arthur, and P.Y.O. Amoako, "Causes of Failure of Students in Computer Programming Courses: The Teacher – Learner Perspective," *Int. J. Comput. Appl. (0975 – 8887)*, vol. 77, no. 12, pp. 27–32, 2013.
- [4] H. Rahim, H.B. Zaman, A. Ahmad, and N.M. Ali, "Student' s Difficulties in Learning Programming," *Adv. J. Tech. Vocat. Educ.*, vol. 2, no. 3, pp. 40–43, 2018.
- [5] F. Mohan, "Teaching C ++ Programming as a foreign language," in *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*, 2018, pp. 18–23.
- [6] Y. Qian and J.D. Lehman, "Students' Misconceptions and Other Difficulties in Introductory Programming: A Literature Review," *ACM Trans. Comput. Educ.*, vol. 18, no. February, 2017.
- [7] A. Edgcomb, F. Vahid, R. Lysecky, and S. Lysecky, "Getting students to earnestly do reading, studying, and homework in an introductory programming class," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education.*, 2017, pp. 171–176.
- [8] N.S. Ibrahim, H. Hasmy, A.Y. Kapi, and A. Suhani, "The Usability of C++ Interactive Self-Assessment Quiz (i-SAQ)," *Int. J. Hum. Technol. Interact.*, vol. 2, no. 2, pp. 67–74, 2018.
- [9] F.L. Khaleel, N.S. Ashaari, and T.S.M.T. Wook, "An Empirical Stusy On Gamification For Learning Programming Language," *J. Teknol.*, vol. 81, no. 2, pp. 151–162, 2019.
- [10] A.L. Asnawi et al., "The Needs of Collaborative Tool for Practicing Pair Programming in Educational Setting," *Int. J. Interact. Mob. Technol.*, no. July, 2019.

